

# **RETAIL CUSTOMER ANALYTICS PROJECT**

## **PROJECT OVERVIEW:**

In this project, Pandas, SQL, and Power BI were used together to build an end-to-end data analytics solution. Pandas was used as the primary tool for data loading, data cleaning, pre-processing, and feature engineering. Real-world datasets often contain missing values, inconsistent column names, and redundant information, which makes direct analysis difficult. Pandas provided the flexibility and efficiency needed to clean, standardize, and validate the customer shopping behaviour dataset before analysis.

Once the data was cleaned and prepared using Pandas, it was stored in a MySQL database. SQL was then used to perform structured data analysis, including filtering, aggregation, grouping, and comparison to answer key business questions. This step helped transform the cleaned data into meaningful insights.

Finally, Power BI was used to visualize the results of the analysis through interactive dashboards. By connecting Power BI directly to the MySQL database, insights were presented using charts, tables, KPI cards, and conditional formatting. This end-to-end approach ensured that data moved seamlessly from raw form to actionable insights, demonstrating a real-world analytics workflow suitable for interview discussions.

## **DATA CLEANING USING PANDAS:**

After loading the dataset, Pandas was used to clean and standardize the data. Missing values were identified and handled carefully to avoid bias in analysis. Instead of using a single global value, category-wise median values were used to fill missing review ratings, which preserves category-level patterns. Column names were standardized by converting them to lowercase and replacing spaces with underscores. This step improved readability and avoided issues while writing SQL queries and creating Power BI visuals. Redundant columns were identified using logical validation checks and removed to keep the dataset clean and efficient.

- `df.isnull().sum()`
- `df["review_rating"] = df.groupby("category")["review_rating"].transform("median")`
- `df.columns = df.columns.str.lower().str.replace(" ", "_")`

## FEATURE ENGINEERING & FINAL DATASET:

Feature engineering was performed using Pandas to improve analytical capabilities. Customers were grouped into age segments using quartile-based binning. Creating such features helps analyse customer behaviour more effectively across different segments. Before moving the data to the database, final validation checks were performed to ensure correctness and remove unnecessary columns. After these steps, the dataset was considered analysis-ready and suitable for structured querying and visualization.

- `lables = ["young adults", "adults", "middle-aged", "senior"]`
- `df["age_group"] = pd.qcut(df["age"], q=4, labels = lables)`

## SQL OVERVIEW:

After data cleaning, the final dataset was stored in a MySQL database. SQL was used mainly for **data analysis and answering business questions**, not for data cleaning. Using SQL allowed structured querying, aggregation, and comparison of data efficiently. SQL queries were written to analyse customer behaviour, category performance, and purchase trends. Aggregation functions such as AVG, SUM, and COUNT were used to calculate key metrics. GROUP BY was used to analyse performance at different levels such as category and age group

### Examples:

```
select item_purchased,
100 * sum(case when discount_applied = "yes" then 1 else 0 end) /
count(*) as high
from customers
group by item_purchased
order by high desc limit 5;
```

Result Grid			Filter Rows:
	item_purchased	high	
▶	Hat	50.0000	
	Sneakers	49.6552	
	Coat	49.0683	
	Sweater	48.1707	
	Pants	47.3684	

```
select case when previous_purchases = 1 then "New",
when previous_purchases <= 10 then "Returning" else "Loyal"
end as segment,
count(previous_purchases)
from customers
group by segment;
```

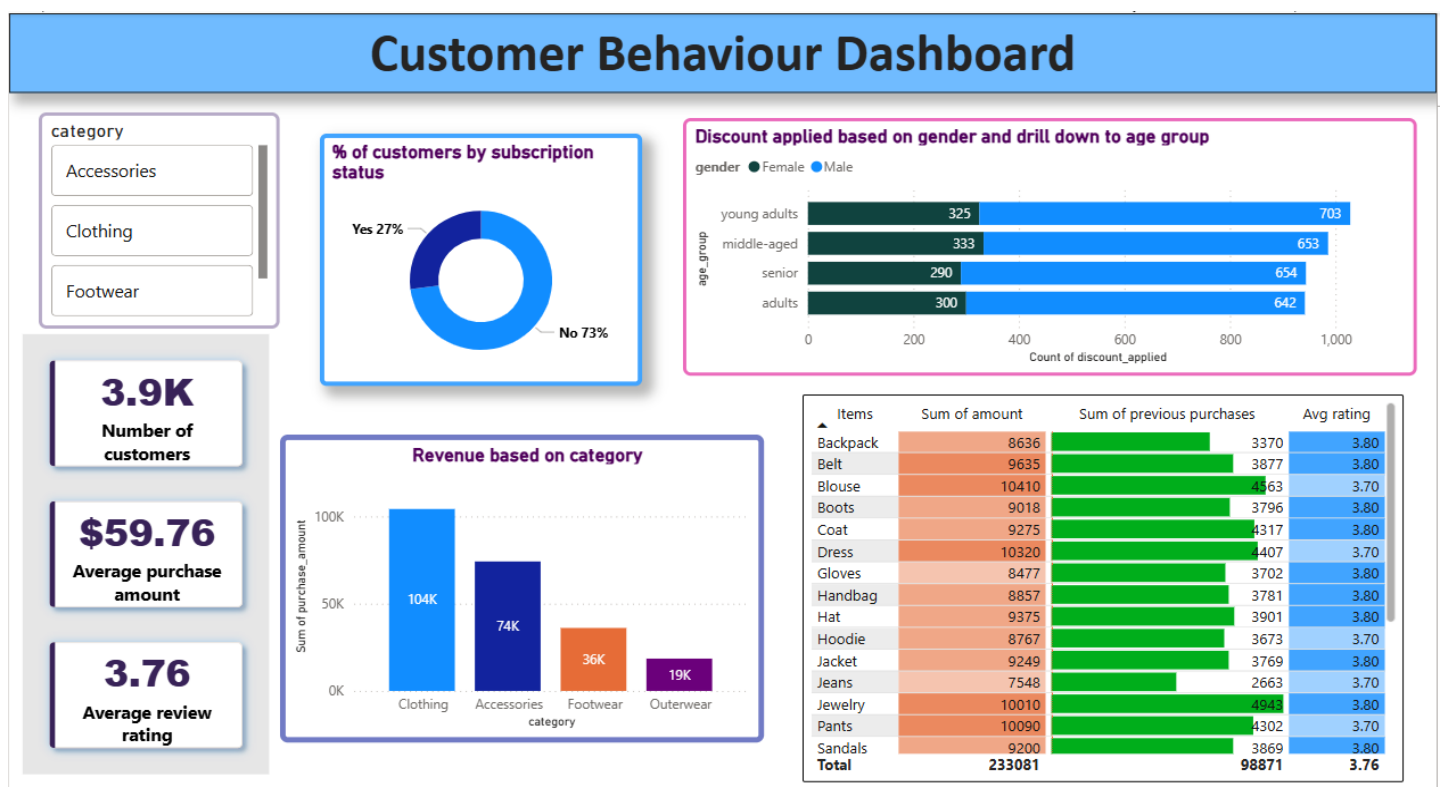
Result Grid			Filter Rows:
	segment	count(previous_purchases)	
▶	Loyal	3116	
	Returning	701	
	New	83	

```
select avg(purchase_amount), shipping_type
from customers
where shipping_type in ("Express", "Standard" )
group by shipping_type;
```

Result Grid			Filter Rows:
	avg(purchase_amount)	shipping_type	
▶	60.4752	Express	
	58.4602	Standard	

## POWER BI OVERVIEW:

Power BI was used as the final step of the project to visualize insights in an interactive and user-friendly dashboard. The data was connected directly from the MySQL database to ensure consistency between analysis and reporting. **KPI cards** were created to display key metrics such as total customers, average purchase amount, and average review rating. Category-wise revenue and subscription status were visualized using bar and donut charts. **Drill-down** functionality was implemented in the discount analysis to explore data by gender and further by age group. **Conditional formatting** was applied to tables to highlight high and low values for quick interpretation. Filters and slicers were added to enable dynamic analysis across different product categories.



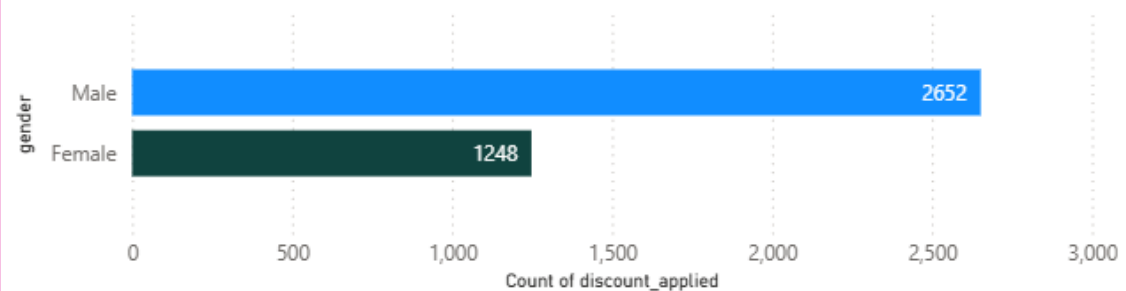
## Conditional Formatting:

Items	Sum of amount	Sum of previous purchases	Avg rating
Backpack	8636	3370	3.80
Belt	9635	3877	3.80
Blouse	10410	4563	3.70
Boots	9018	3796	3.80
Coat	9275	4317	3.80
Dress	10320	4407	3.70
Gloves	8477	3702	3.80
Handbag	8857	3781	3.80
Hat	9375	3901	3.80
Hoodie	8767	3673	3.70
Jacket	9249	3769	3.80
Jeans	7548	2663	3.70
Jewelry	10010	4943	3.80
Pants	10090	4302	3.70
Sandals	9200	3869	3.80
<b>Total</b>	<b>233081</b>	<b>98871</b>	<b>3.76</b>

## Drill Down:

Discount applied based on gender and drill down to age group

gender ● Male ● Female



Discount applied based on gender and drill down to age group

gender ● Female ● Male

